

量子化信息素蚁群优化特征选择算法

李占山^{1,2}, 刘兆赓², 俞寅², 鄢文浩²
(1. 吉林大学 计算机科学与技术学院, 吉林 长春 130012; 2. 吉林大学 软件学院, 吉林 长春 130012)

摘 要: 蚁群优化算法凭借其正反馈机制和强大的搜索能力被广泛地应用于各类优化问题求解上. 本文试图将蚁群优化算法应用于特征选择领域并提出了新的量子化信息素蚁群优化(quantized pheromone ant colony optimization, QPACO)特征选择算法. 相比于其他基于蚁群优化算法的特征选择算法, QPACO 算法中采用了量子化信息素的启发式策略, 改变了传统的信息素更新策略, 因此避免了在搜索特征时的局部最优问题. 实验采用了 KNN 分类器来指导学习过程, 利用源于 UCI 数据库的多组数据集进行了相关的测试, 实验结果表明, QPACO 算法在分类精度、精确率、召回率和维度缩减率等方面均具有良好的性能.

关 键 词: 特征选择; 蚁群优化; 信息素; 量子化; 启发式策略

中图分类号: TP 18 文献标志码: A 文章编号: 1005-3026(2020)01-0017-06

A Quantized Pheromone Ant Colony Optimization Algorithm for Feature Selection

LI Zhan-shan^{1,2}, LIU Zhao-geng², YU Yin², YAN Wen-hao²
(1. College of Computer Science and Technology, Jilin University, Changchun 130012, China; 2. College of Software, Jilin University, Changchun 130012, China. Corresponding author: YU Yin, E-mail: 102792556@qq.com)

Abstract: Ant colony optimization algorithms have positive feedback mechanisms and strong searching abilities, which makes them widely used in various kinds of optimization problems. An ant colony optimization algorithm was applied to the field of feature selection and a new quantized pheromone ant colony optimization(QPACO) feature selection algorithm was proposed. Quantum pheromone heuristic strategy was adopted in QPACO algorithm, compared with other ant colony optimization algorithms for feature selection, QPACO algorithm changes the traditional pheromone updating strategy and avoids the local optimization problem when searching for features. In the experimental stage, a KNN classifier was used to guide the learning process, and multiple data sets from the UCI database were used for testing. The experimental results showed that QPACO algorithm has good performances in classification accuracy, precision, recall and feature-reduction.

Key words: feature selection; ant colony optimization; pheromone; quantization; heuristic strategy

近年来,许多涉及信息的领域中产生了包含众多特征的高维数据,然而并不是所有特征都是重要的.许多特征甚至是不相关或冗余的,这不仅使数据处理过程变得困难,还降低了学习算法的效率,如分类算法等学习算法的性能^[1].特征选择旨在利用一种选择策略,消除不相关和冗余的特征,找到最佳特征子集^[2].特征选择是一个复杂的问题,对于一个有 n 个特征的数据集,可能的解决方案的总数是 $2^n - 1$ ^[3],其搜索空间十分庞大.因此,用穷举搜索选择一组最优特征的时间复杂度是 $O(2^n)$ ^[4],这在大多数情况下是不切实际的.与传统方法相比,基于演化算法的特征选择方法更适合于解决这种问题.

本文设计了一种基于随机二元蚁群网络的特

征选择方法,更换了启发式因子的计算方法,并提出了一种新的信息素更新思路,将整体的信息素量子化,赋予每个信息素生命周期,形成了量子化信息素蚁群优化(quantized pheromone ant colony optimization, QPACO)特征选择算法。

1 相关工作

近年来,基于演化计算的特征选择算法受到了广泛研究,基于演化计算的特征选择算法已发表论文超过 500 篇^[1]。

1.1 演化计算用于特征选择

基于演化计算的特征选择算法的研究近年来取得了较为令人满意的结果。如 Rashedi 等提出了通过增强传递函数克服停滞问题的 IBGSA^[5], IBGSA 将二进制向量每个位与一个特征相关联,通过寻找最优二进制向量找到特征选择的最优解;Chuang 等在二元粒子群算法 BPSO 中引入鲶鱼效应,提出了 Catfish BPSO^[6], Catfish BPSO 将局部最优中的粒子通过鲶鱼效应引导到新的搜索空间,同时使用种群中最差的适应值替换 10% 的原始粒子,最终避免了局部最优,进一步获得了更好的解;初蓓等提出了改进的森林优化特征选择算法 IFSFOA^[7],采用了新的初始化策略和更新机制进一步提高原始算法的性能。

1.2 蚁群优化算法用于特征选择

本文主要讨论基于蚁群优化算法的特征选择算法。蚁群优化(ant colony optimization, ACO)算法是 Dorigo 等提出的一种演化算法^[8]。蚂蚁之间的通信会产生正反馈行为,引导蚁群收敛到最优解。蚁群路径上分布的信息素模拟了真实蚂蚁所经过的路线上的化学物质^[9]。

基于蚁群优化算法解决特征选择的主要思想是将问题建模为求解搜索图的最小成本路径问题,创建一个包含节点的搜索空间并设计一个程序来寻找解决方案的路径,蚂蚁的路径表示特征的子集。ACO 算法基于信息素和启发式信息计算蚂蚁选择解路径的转移概率。Chen 等使用这种类型的 ACO 算法进行特征选择,提出了 ACOFS^[10],ACOFS 中使用了 F_score 标准作为启发式值,但采用了不同的信息素更新策略;Kashef 等提出了一种优化的二元蚁群算法 ABACO^[11],该算法的不同之处在于每个特征都有两个节点,一个节点用于选择,另一个节点用于取消选择相应的特征,最优特征子集是满足遍历停止条件的最小节点数的蚂蚁遍历图。

与上述特征选择算法相比,本文提出的 QPACO 算法,采用了新的启发式因子的计算方法,改变了传统的信息素的更新策略,避免了搜索特征时的局部最优,提高了特征选择的精度。

2 QPACO 算法

2.1 基于蚁群优化算法的特征选择算法

基于蚁群优化算法的特征选择算法首先构造了由 n 个特征组成的有向图,假设蚂蚁在初始时刻被随机放置在 n 个特征节点中,并维护一张禁忌列表记录每只蚂蚁已经访问过的节点,每条边的信息素 $\tau_{i,j}$ 初始值为 0,蚂蚁依据边上的信息素计算在 t 次迭代时,蚂蚁 k 从特征 i 移动到特征 j 的概率:

$$p_{i,j}^k(t) = \begin{cases} \frac{\tau_{i,j}^\alpha(t) \eta_{i,j}^\beta(t)}{\sum_{g \notin S} \tau_{i,g}^\alpha(t) \eta_{i,g}^\beta(t)}, & j \notin S; \\ 0, & \text{否则}. \end{cases} \quad (1)$$

式中: S 是蚂蚁 k 的禁忌列表; α 是信息启发因子; β 是期望启发式因子,用于分配启发式信息和信息素浓度的权重; $\eta_{i,j}$ 是启发式信息,通常计算为

$$\eta_{i,j} = \frac{1}{d_{i,j}}. \quad (2)$$

式中, $d_{i,j}$ 是特征 i 与特征 j 之间的欧几里得度量。

当蚂蚁完成一次迭代后,每条路径上的信息素都会进行更新:

$$\tau_{i,j} \leftarrow (1-p) \tau_{i,j} + p \sum_{k=1}^m \Delta \tau_{i,j}^k. \quad (3)$$

式中: p 为权重系数 ($0 < p < 1$); m 为蚂蚁总数; $\Delta \tau_{i,j}^k$ 为蚂蚁 k 在此次迭代时特征 i 到特征 j 之间的信息素增量,其计算如下:

$$\Delta \tau_{i,j}^k = \begin{cases} \frac{Q}{L_k}, & \text{蚂蚁 } k \text{ 访问特征 } i \text{ 到特征 } j; \\ 0, & \text{否则}. \end{cases} \quad (4)$$

式中: Q 为常数; L_k 为蚂蚁 k 在此次迭代中的路径长度。

基于蚁群优化算法的特征选择算法在特征选择领域上有一定的作用,但仍存在一些不足,因此提出了许多不同版本的改进。

2.2 二元蚁群优化特征选择算法

二元蚁群优化(binary ant colony optimization, BACO)特征选择算法是另一种常用的求解算法。该方法仅允许全局最优蚂蚁存放信息素,并采用最大最小策略进行信息素更新,即信息素轨迹被

限制在 $[\min, \max]$ 区间内,其中 \min 和 \max 是满足 $\min < \max$ 的任意正实数.这里,节点表示特征,其中每个节点包含两个子节点:0和1.首先,所有蚂蚁都在比特串的开始节点,然后通过选择不同的子节点爬向终止节点.如果蚂蚁选择第 i 个特征的子节点为1或0,则代表着该特征被该蚂蚁选择或未选择.假设在迭代中0到1(0 \rightarrow 1)之间的子路径的优选概率被计算,计算公式为

$$p_{01}(t) = \frac{\tau_{01}}{\tau_{00} + \tau_{01}}. \quad (5)$$

式中: p_{01} 是与子路径(0 \rightarrow 1)相关联的概率; τ_{00} 和 τ_{01} 是子路径(0 \rightarrow 0和0 \rightarrow 1)上的信息素.

虽然BACO能够在短时间内找到近似最优解,但是它依旧面临着一些问题,如蚂蚁对特征的看法有限及特征选择的准确率有待提高.在本文提出的算法中,尝试结合传统的ACO和BACO来解决上述问题.

2.3 量子化蚁群优化特征选择算法

鉴于现有算法的准确率仍待提高,经过研究,本文提出了量子化信息素蚁群优化特征选择算法QPACO.

2.3.1 路径转移概率计算

蚂蚁在特征图中的转移按照概率进行,在QPACO算法中,每一个特征均包含其子节点1或0,则意味着该特征被该蚂蚁选择或未选择.那么在特征 i 与 j 之间,显然存在4条路径(0 \rightarrow 0, 0 \rightarrow 1, 1 \rightarrow 0, 1 \rightarrow 1).这些路径上的转移概率计算方式如下:

$$p_{i_x, j_y}^k(t) = \begin{cases} \frac{\tau_{i_x, j_y}^\alpha \eta_{i_x, j_y}^\beta}{\sum_{k \in C} \tau_{i_x, k_0}^\alpha \eta_{i_x, k_0}^\beta + \sum_{k \in C} \tau_{i_x, k_1}^\alpha \eta_{i_x, k_1}^\beta} & j \in C; \\ 0, & \text{否则.} \end{cases} \quad (6)$$

式中: i_x, j_y 分别表示特征 i 和 j 中的子节点; C 表示蚂蚁能够访问且尚未被访问到的节点集合; τ 表示子节点间的信息素; η 表示子节点间的启发式信息; α 和 β 是确定信息素和启发信息值的两个参数.

2.3.2 信息素 τ 的计算与更新

在一般的蚁群和蚁群变种的特征选择算法中,通常人们采用设定一个常量 q ($0 < q < 1$,一般为0.6~0.8)来模拟信息素的消失:

$$\tau_{\text{new}} = \tau_{\text{old}} \times q + \Delta\tau. \quad (7)$$

式中: q 为信息素模拟消失常量; τ_{old} 为原来的信息素值; τ_{new} 为更新后的信息素值; $\Delta\tau$ 为信息素变化量.

QPACO算法中提出了一种新的信息素挥发方式,即把信息素量子化,看成是一份一份有着时

间标记的信息素单元,每一单元信息素都有着属于自己的生命周期,当自己生命周期结束时挥发,其简化公式如下:

$$\tau_{\text{new}} = \tau_{\text{old}} - \tau_{\text{dead}} + \Delta\tau. \quad (8)$$

式中, τ_{dead} 为在该次迭代中生命周期结束的信息素量.

考虑到信息素值的上下限问题,因此在 k 次迭代时信息素的更新 τ_{new} 和信息素更新量的记录 $\Delta\tau_k$ 计算如下:

$$\tau_{\text{temp}} = \begin{cases} \tau_{\text{old}} - \Delta\tau_{k-h}, & \tau_{\text{old}} - \Delta\tau_{k-h} > \tau_{\min}; \\ \tau_{\min}, & \tau_{\text{old}} - \Delta\tau_{k-h} \leq \tau_{\min}; \end{cases}$$

$$\tau_{\text{new}} = \begin{cases} \tau_{\text{temp}} + \sum_m \Delta\tau, & \tau_{\text{temp}} + \sum_m \Delta\tau < \tau_{\max}; \\ \tau_{\max}, & \tau_{\text{temp}} + \sum_m \Delta\tau \geq \tau_{\max}; \end{cases}$$

$$\Delta\tau_k = \tau_{\text{new}} - \tau_{\text{temp}}. \quad (9)$$

式中: $\sum_m \Delta\tau$ 为该条路径上所有蚂蚁所留下的信息素总和; $\Delta\tau_k$ 为该条路径在本次迭代(k 次迭代)时的更新量; $\Delta\tau_{k-h}$ 为 h 次迭代前该条路径上的信息素更新量.需注意这三者间的区分, h 变量即为最初所说的信息素单元的生命周期.

2.3.3 启发式信息 η 的计算

在参考了众多 η 的计算方法后,发现基于 F_{score} 的改进启发式度量方式更适用于本文的算法.

F_{score} 是用于评价特征识别能力的度量,第 i 个特征的 F_{score} 公式为

$$F_{\text{score}_i} = \frac{\sum_{t=1}^c (\bar{x}_i^t - \bar{x}_i)^2}{\sum_{s=1}^c [(1/(N_i^s - 1)) \sum_{u=1}^{N_i^s} (x_{i,u}^s - \bar{x}_i^s)^2]}. \quad (10)$$

式中: c 为类别的数目; \bar{x}_i^t 为类别 t ($t=1, 2, \dots, c$)中特征 i ($i=1, 2, \dots, n$)的平均值; \bar{x}_i 为所有类别中特征 i ($i=1, 2, \dots, n$)的平均值; n 为特征的数目; N_i^s 为类别 s ($s=1, 2, \dots, c$)中特征 i ($i=1, 2, \dots, n$)的样本数; $x_{i,u}^s$ 为类别 s ($s=1, 2, \dots, c$)中特征 i ($i=1, 2, \dots, n$)的第 u ($u=1, 2, \dots, N_i^s$)次训练样本数; \bar{x}_i^s 为类别 s ($s=1, 2, \dots, c$)中特征 i ($i=1, 2, \dots, n$)的平均值.

越大的 F_{score} 意味着该特征具有判别力的可能性越大^[12],使用该标准的边的启发信息计算如下:

$$\left. \begin{aligned} \eta_{i_0, j_0} &= (\xi/n) \sum_{k=1}^n F_{\text{score}_k}, \\ \eta_{i_0, j_1} &= F_{\text{score}_j}, \\ \eta_{i_1, j_0} &= (\xi/n) \sum_{k=1}^n F_{\text{score}_k}, \\ \eta_{i_1, j_1} &= F_{\text{score}_j}. \end{aligned} \right\} \quad (11)$$

式中: η_{i_x,j_y} 代表特征 i 取值为 x 时与特征 j 取值为 y 时所构成的边上的启发式信息(x 和 y 的取值均为 0 或 1); n 为特征的数目; ξ 为常量(通常取值在 0~1 之间)。

2.4 QPACO 算法伪代码

QPACO 算法伪代码见表 1。

表 1 QPACO 算法伪代码

Table 1 Pseudo code of QPACO algorithm

输入:数据集、数据集类别、迭代次数 (dataset, dataclass, $t_{iteration}$)
输出:拥有最高适应度的最优特征子集
1. Procedure QPACO
2. 检验读入数据
3. 初始化 $\alpha, \beta, T_0 = 0, \min T, \max T$ (信息素值的上下限)
4. 置信息素 τ 的初值为 T_0
5. for $i = 1$ to $t_{iteration}$ do
6. 运用式(10)和式(11)计算启发式信息
7. 运用式(6)计算每只蚂蚁在每条路上的选择概率
8. 运用式(9)更新信息素 τ
9. 记录 τ 的更新量 $\Delta\tau$
10. 评估构造子集并选择局部最优和全局最优子集
11. End for
12. 返回拥有最高适应度的最优特征子集

3 实验结果与分析

3.1 实验数据与对比算法

本文从 UCI 数据库中选择了一些典型的数据集对算法进行验证。典型的数据集见表 2。

表 2 UCI 数据集

Table 2 UCI datasets

数据集	分类数	特征数	样本数
Glass	6	9	214
Letter	26	16	20 000
Shuttle	7	9	58 000
Spambase	2	57	4 601
Vehicle	4	18	846
Waveform	3	21	5 000
Wine	3	13	178
Wisconsin	2	9	683
Yeast	9	8	1 484

为了更好地展现算法的可比性,选择了一些具有代表性的特征选择算法与本文算法进行比较,其中包括 Catfish BPSO^[6]、改进二元引力搜索(IBGSA)^[5]、基于蚁群优化算法的特征选择算法 ACOFS^[10]和 ABACO_H^[13]、改进的森林优化特征选择算法 IFSFOA^[7]和二元蝴蝶优化特征选择算法 S-bBOA^[14]等。

3.2 实验环境、评估指标及实验参数

3.2.1 实验环境

实验中使用了 python 3.6 实现了本文的算法,同时使用了公开的工具包 scikit-learn。所有实验均在一台配置为 Intel Core i5-4210H (CPU),8GB 内存、1TB 硬盘的计算机上完成。

3.2.2 评估指标

在本实验中,使用分类精度(A_c)、精确率(R_p)、召回率(R_{recall})和维度缩减率(R_f)来评估所提出的算法性能。

分类精度(A_c),即正确分类的样本数和测试集的总样本数的百分比,其定义为

分类精度 = $\frac{\text{正确分类的样本数}}{\text{测试集的总样本数}}$

(12)

精确率(R_p)和召回率(R_{recall})计算式如下:

$$R_p = \frac{TP_i}{TP_i + FP_i},$$

(13)

$$R_{recall} = \frac{TP_i}{TP_i + FN_i}.$$

(14)

其中: TP_i 为第 i 类下正确分类的测试样本数; FP_i 为第 i 类下错误分类的测试样本数; FN_i 为在其他类别下错误分类的测试样本数。

定义维度缩减率(R_f)如下:

$$R_f = \frac{n-p}{n}.$$

(15)

其中: n 是总特征数; p 是算法所选择的特征数。

3.2.3 算法参数设置

在 QPACO 算法与其他算法的对比实验中,统一设置了算法的一些参数:所有算法的种群数量和迭代次数为 50;在每次实验中,60% 的样本随机选择进行训练,剩下 40% 的样本用于测试;实验结果在每个数据集和算法中独立运行超过 20 次,最后统计平均值;挥发系数 ρ 为 0.049;每条路径上的最小和最大信息素强度分别设定为 0.1 和 6;每个边缘的初始信息素强度设定为 0.1; α 和 β 是决定信息素和启发信息的相对重要性的两个参数,设 $\alpha = 1, \beta = 0.5$;在分类器的选择上,使用了 K 近邻(KNN)分类器作为基分类器,并将参数 K 设置为 1。

3.3 实验结果与分析

3.3.1 实验结果

为了确保对比实验的准确性,表 3 与表 4 中的部分数据采用了文献[13]中的实验结果,表 3 是 QPACO 算法与其对比算法在不同数据集上的分类精度的对比。表 4 是 QPACO 算法与其对比算法在不同数据集上的精确率、召回率及维度缩减率

的对比,最后一列为每个算法在所有数据集上的指标和,和越大说明算法总体性能越好.

表 3 QPACO 及其对比算法的平均分类精度对比

Table 3 Comparison of average classification accuracy about QPACO and its comparison algorithms									
数据集	QPACO	ABACO _H	ABACO	ACOFS	BACO	IBGSA	Catfish BPSO	IFSFOA	S – bBOA
Glass	0.771 (1)	0.758 (2)	0.747 (5)	0.75 (4)	0.743 (6)	0.738 (7)	0.733 (8)	0.721 (9)	0.756 (3)
Letter	0.962 (1)	0.861 (4)	0.856 (8)	0.859 (6)	0.859 (6)	0.86 (5)	0.856 (8)	0.95 (3)	0.959 (2)
Shuttle	0.999 (1)	0.998 (3)	0.998 (3)	0.998 (3)	0.998 (3)	0.998 (3)	0.998 (3)	0.999 (1)	0.998 (3)
Spambase	0.929 (1)	0.923 (3)	0.921 (7)	0.922 (5)	0.919 (8)	0.922 (5)	0.924 (2)	0.918 (9)	0.923 (3)
Vehicle	0.74 (7)	0.753 (1)	0.753 (1)	0.749 (3)	0.749 (3)	0.745 (6)	0.746 (5)	0.74 (7)	0.737 (9)
Waveform	0.802 (1)	0.798 (2)	0.795 (5)	0.797 (4)	0.793 (8)	0.794 (7)	0.79 (9)	0.798 (2)	0.795 (5)
Wine	0.976 (2)	0.969 (4)	0.969 (4)	0.964 (7)	0.964 (7)	0.978 (1)	0.969 (4)	0.944 (9)	0.972 (3)
Wisconsin	0.978 (1)	0.976 (3)	0.976 (3)	0.974 (7)	0.975 (6)	0.977 (2)	0.976 (3)	0.964 (9)	0.967 (8)
Yeast	0.525 (2)	0.524 (3)	0.529 (1)	0.516 (5)	0.515 (6)	0.515 (6)	0.522 (4)	0.483 (9)	0.51 (8)

注:括号中的数字表示在各个数据集上每种算法性能的排名.

3.3.2 实验分析

QPACO 算法在时间效率上是基本等同于二元蚁群优化算法的,在算法的改进过程中,计算和记录每次信息素的更新量,以及查找生命周期结束信息素量的时间复杂度都是 $O(1)$,因此时间上的开销差距并不明显,所以本文遵从了相关的基于演化计算的特征选择方法的实验设计模式,并未采用时间效率来衡量算法性能^[13-17],但计算了其他常用的评估指标进行算法间的对比.

对比分类精度,从表 3 中不难看出,QPACO 算法在 Glass, Letter, Shuttle, Spambase, Waveform, Wisconsin 这些数据集上均位居第一,在 Wine 和 Yeast 上位居第二,只在 Vehicle 上稍显逊色.因此 QPACO 算法在分类精度上有了很明显的提升.通过对比表 4 中的精确率、召回率以及维度缩减率,发现 QPACO 算法大多数情况下精确率和召回率都略高于其他算法,且维度缩减率维持在平均水平以上.

表 4 QPACO 及其对比算法的精确率、召回率、维度缩减率对比

Table 4 Comparison of precision, recall and feature-reduction about QPACO and its comparison algorithms											
算法	指标	Glass	Letter	Shuttle	Spambase	Vehicle	Waveform	Wine	Wisconsin	Yeast	Sum
QPACO	R_p	0.728	0.962	0.913	0.927	0.738	0.802	0.978	0.976	0.474	7.498
	R_{recall}	0.735	0.962	0.912	0.925	0.742	0.802	0.979	0.977	0.469	7.503
	R_f	0.289	0.313	0.489	0.459	0.422	0.276	0.477	0.333	0.175	3.233
ABACO _H	R_p	0.743	0.876	0.912	0.92	0.755	0.799	0.972	0.973	0.529	7.479
	R_{recall}	0.736	0.862	0.927	0.92	0.757	0.799	0.973	0.976	0.521	7.471
	R_f	0.299	0.343	0.541	0.431	0.436	0.585	0.484	0.384	0.063	3.566
ABACO	R_p	0.693	0.863	0.912	0.919	0.755	0.797	0.97	0.973	0.504	7.386
	R_{recall}	0.676	0.861	0.907	0.916	0.762	0.797	0.973	0.976	0.510	7.378
	R_f	0.361	0.335	0.453	0.439	0.428	0.272	0.549	0.348	0.076	3.261
ACOFS	R_p	0.723	0.864	0.958	0.921	0.749	0.798	0.965	0.969	0.509	7.456
	R_{recall}	0.71	0.861	0.934	0.918	0.752	0.798	0.97	0.974	0.496	7.413
	R_f	0.365	0.361	0.522	0.447	0.426	0.272	0.483	0.367	0.102	3.345
BACO	R_p	0.73	0.865	0.944	0.917	0.749	0.794	0.965	0.973	0.509	7.446
	R_{recall}	0.695	0.863	0.93	0.917	0.754	0.794	0.972	0.974	0.507	7.406
	R_f	0.347	0.332	0.565	0.465	0.439	0.261	0.494	0.387	0.089	3.379
IBGSA	R_p	0.672	0.84	0.911	0.906	0.728	0.773	0.93	0.966	0.472	7.198
	R_{recall}	0.668	0.835	0.928	0.905	0.732	0.773	0.935	0.946	0.474	7.196
	R_f	0.315	0.42	0.381	0.465	0.413	0.51	0.341	0.386	0.129	3.36
Catfish BPSO	R_p	0.706	0.865	0.867	0.928	0.749	0.792	0.972	0.976	0.496	7.351
	R_{recall}	0.699	0.862	0.935	0.926	0.754	0.793	0.974	0.977	0.491	7.411
	R_f	0.359	0.34	0.527	0.443	0.451	0.305	0.513	0.392	0.134	3.464
IFSFOA	R_p	0.717	0.95	0.917	0.914	0.731	0.797	0.94	0.959	0.512	7.437
	R_{recall}	0.729	0.95	0.876	0.913	0.736	0.797	0.957	0.963	0.529	7.45
	R_f	0.444	0.375	0.556	0.421	0.389	0.381	0.692	0.556	0.125	3.939
S – bBOA	R_p	0.803	0.959	0.77	0.924	0.732	0.795	0.979	0.961	0.401	7.324
	R_{recall}	0.733	0.959	0.677	0.915	0.735	0.795	0.967	0.968	0.383	7.132
	R_f	0.556	0.312	0.778	0.456	0.611	0.333	0.615	0.667	0.25	4.578

4 结 论

本文提出了基于传统蚁群优化算法和二元蚁群优化算法的量子化信息素蚁群优化(QPACO)特征选择算法。QPACO 算法中将信息素量子化,赋予每个信息素单独的生命周期,同时修改了启发式因子的更新方式,增强了算法的搜索能力。经过 9 个数据集和 9 个特征选择算法的对比实验,验证了 QPACO 算法良好的性能。如何在高维数据集中应用 QPACO 算法进行特征选择问题的求解,将是下一步重点研究的内容。

参考文献:

- [1] Xue B, Zhang M, Browne W N, et al. A survey on evolutionary computation approaches to feature selection[J]. *IEEE Transactions on Evolutionary Computation*, 2016, 20(4):606–626.
- [2] Zhang Z, Bai L, Liang Y, et al. Joint hypergraph learning and sparse regression for feature selection [J]. *Pattern Recognition*, 2017, 63:291–309.
- [3] Guyon I, Elisseeff A. An introduction to variable and feature selection[J]. *Journal of Machine Learning Research*, 2003, 3(6):1157–1182.
- [4] Tan K C, Teoh E J, Yu Q, et al. A hybrid evolutionary algorithm for attribute selection in data mining[J]. *Expert Systems with Applications*, 2009, 36(4):8616–8630.
- [5] Rashedi E, Nezamabadi-pour H. Feature subset selection using improved binary gravitational search algorithm[J]. *Journal of Intelligent & Fuzzy Systems*, 2014, 26(3):1211–1221.
- [6] Chuang L Y, Tsai S W, Yang C H. Improved binary particle swarm optimization using catfish effect for feature selection [J]. *Expert Systems with Applications*, 2011, 38(10):12699–12707.
- [7] 初蓓,李占山,张梦林,等. 基于森林优化特征选择算法的改进研究[J]. *软件学报*, 2018, 29(9):2547–2558.
(Chu Bei, Li Zhan-shan, Zhang Meng-lin, et al. Research on improvements of feature selection using forest optimization algorithm[J]. *Journal of Software*, 2018, 29(9):2547–2558.)
- [8] Dorigo M, Caro G D. Ant colony optimization: a new meta-heuristic [C]//Proceedings of the 1999 Congress on Evolutionary Computation. Washington D C, 1999:1470–1477.
- [9] Dorigo M, Birattari M, Stützle T. Ant colony optimization: artificial ants as a computational intelligence technique[J]. *IEEE Computational Intelligence Magazine*, 2006, 1(4):28–39.
- [10] Chen B, Chen L, Chen Y. Efficient ant colony optimization for image feature selection [J]. *Signal Processing*, 2013, 93(6):1566–1576.
- [11] Kashef S, Nezamabadi-Pour H. A new feature selection algorithm based on binary ant colony optimization[C]//The 5th Conference on Information and Knowledge Technology. Shiraz, Iran, 2013:50–54.
- [12] Huang C L. ACO-based hybrid classification system with feature subset selection and model parameters optimization [J]. *Neurocomputing*, 2009, 73(1/2/3):438–448.
- [13] Kashef S, Nezamabadi-Pour H. An advanced ACO algorithm for feature subset selection [J]. *Neurocomputing*, 2015, 147:271–279.
- [14] Arora S, Anand P. Binary butterfly optimization approaches for feature selection [J]. *Expert Systems with Applications*, 2019, 116:147–160.
- [15] Mafarja M M, Mirjalili S. Hybrid whale optimization algorithm with simulated annealing for feature selection[J]. *Neurocomputing*, 2017, 260:302–312.
- [16] Emary E, Zawbaa H M, Hassanien A E. Binary grey wolf optimization approaches for feature selection [J]. *Neurocomputing*, 2016, 172:371–381.
- [17] Zhang Y, Song X, Gong D. A return-cost-based binary firefly algorithm for feature selection [J]. *Information Sciences*, 2017, 418/419:561–574.